Design and Implementation of Socket Programming through an Android Application.

Rishabh Bhardwaj
201301431

Stuti Mohgaonkar
201301185

Rashmi Wilson
201301033

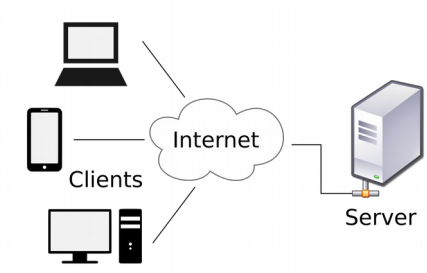
***Abstract*—This paper explains how we made an android application, and used it as a remote mouse to control the cursor and keyboard of a computer, using the concepts of socket programming. Sockets play an important role in client server applications. Client and server can communicate with each other by writing into or reading from these sockets. We have utilized the robot class in java as well as the connection oriented TCP protocol to establish the connection between server and client to make our application.**

Keywords—sockets;TCP; socket programming; java

I. Introduction

A. Client-Server Communication

In Computer Networking , the computers and other devices connected to the Internet are often referred to as end systems or hosts. Hosts are further divided into two categories : clients and servers. Generally clients consists of individual devices like phones, tablets or PC, while servers run on more powerful machines that store and distribute Web pages, stream videos, and so on. In Client-Server applications clients and servers communicate with each other either from different computers over a network or on the same computer. The client process always initiates the connection to the server, while the server process waits for a connection request from any client. When both the server and client process run on the same computer it is called single seat setup. If the hosts act both as a server and a client, then such an architecture is called peer to peer(P2P).[1]



B. Socket Interface

Two processes communicate with each other by writing into and reading from a software interface called socket. From the Oracle Java documentation the definition of a socket is as follows :

A socket is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to.

Operations of socket:
Sockets perform the following four functions ,

- To connect to a remote machine,
- Send data,
- Receive data and,
- Close the connection.

Socket is the connecting interface between the application layer and the transport layer. A socket is defined jointly from the IP address and the specified port number.[2]

C. Ports

Ports are the endpoints of the communication between processes and they can be identified with specific services. A port is a 16-bit number, used by the host-to-host protocol to identify to which higher-level protocol or application program (process) it must deliver incoming messages. Many well-known port numbers are already reserved by convention to identify with certain specific services on a host. Generally ports are used by transport layer protocols such as Transmission Control Protocol(TCP) and the User Datagram Protocol(UDP). [3]

II. Network Programming with Sockets

To allow communication between the client socket and server socket both the sides run programs that binds a socket to

it. It is this program that facilitates the server-client connection and writing/reading, to/from the socket. The client program creates a socket and attempts to connect that socket to the server. After the connection is made, the server creates a socket object on its end of the communication. The client and server can now communicate by writing to and reading from the socket. A large number of programming languages have in-built libraries that evoke and control sockets.

Figure 2.2.1 and 2.1.2 show the sequence of functions that are called by the server and client side to allow connection and exchange of data.

- The server calls the ServerSocket object, and attaches the socket to the port on which communication is meant to take place
- The Server then listens for client sockets that want to connect to it. it does this through the accept() method.
- While the server is waiting, a client instantiates a Socket object, specifying the server name and port number to connect to.
- The constructor of the socket() class tries to connect the socket to the specified address and port number. If successful then this connection allows the client and the server to communicate.
- On the server side, the accept() method returns a reference to a new socket on the server that is connected to the client's socket.

```
socket()
    |
bind()
    |
recvfrom()
    |
(wait for a sendto request from some client)
    |
(process the sendto request)
    |
sendto (in reply to the request from the client...for example, send an HTML file)
```

Fig 2.1.1 Sequence of functions evoked by the server.

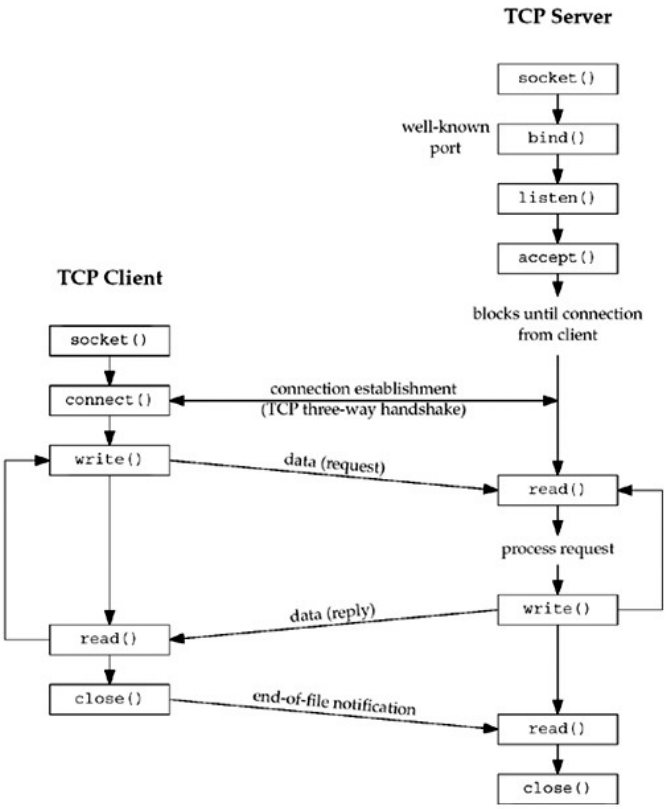
```
socket()
    |
bind()
    |
sendto()
    |
recvfrom()
```

Fig 2.1.2 Sequence of functions evoked by the client.

A. Socket Programming with TCP

Socket Programming can be carried out either over TCP or UDP. For our project we will be using socket programming with TCP.

Transmission Control Protocol(TCP) is a connection oriented service, which means that the connection is first established between the processes. Also when a process sends a packet, it waits for the acknowledgement from the receiver side before sending the next packet. If it does not receive the acknowledgement, then it retransmits the same packet and waits for a longer period of time.



B. Socket Programming over TCP in Java

Java provides a set of classes which are defined in a package called java.net. We make use of this package for socket programming.The two key classes from the java.net package used in creation of server and client programs are:

1. ServerSocket 2. Socket

These functions attempt to create server sockets bound to a particular port. Both the client and Server have socket Objects hence these functions can be used by both the client and server. [4]

The steps for creating a simple server program along with the function required for each step are:

- Open the Server Socket:

```
ServerSocket server = new
ServerSocket( PORT );
```

- Wait for the Client Request:

```
Socket client = server.accept();
```

- Create I/O streams for communicating to the client

```
DataInputStream is = new
DataInputStream(client.getInputStream());
DataOutputStream os = new
DataOutputStream(client.getOutputStream());
```

- Perform communication with client

```
Receive from client: String line =
is.readLine();
```

```
Send to client: os.writeBytes("Hello\n");
```

- Close socket:

```
client.close();
```

C. *Java Robot Class*

According to Sun

"This(Java Robot class) class is used to generate native system input events for the purposes of test automation, self-running demos, and other applications where control of the mouse and keyboard is needed. The primary purpose of Robot is to facilitate automated testing of Java platform implementations."

The java robot class allows your program to take control of your keyboard and mouse. Hence all the actions that are performed by your mouse and keyboard can be carried out by the program till the time it is running. The major functions that are carried out by this class are press key, release key, move mouse, press mouse button, and release mouse button.

The java.awt.Robot library allows the programmer to use the methods of the java robot class.[5]

Some methods of this class include-

- Robot()

  Creates a robot object in the coordinate system of the primary screen.

- keyPress(int keycode)

  Presses the given key on the keyboard.

- keyRelease(int keycode)

  Releases the specified key on the keyboard..

- mouseMove(int x,int y)

  Move the mouse to the given coordinates i.e (x,y)

- mousePress(int buttons)

  Press one or more mouse buttons.

- mouseRelease(int buttons)

  Release one or more mouse buttons.

II. METHODOLOGY

We have undertaken our project in the following two phases:

A. *Phase One*

As per our phase one presentation we have implemented the following :

- Implementation of the Robot Class in Java to control a computer's mouse and keyboard.
- Connecting different laptops as client/server using socket programming.
- Controlling mouse and keyboard of one computer using the other computer.

Since we were testing over two laptops connected over WiFi, the range of the connection between the server and the client depended on the strength of the WiFi.

Ideally, if we know the IP of any laptop connected to the internet, and it is running our code, we can control its mouse, sitting anywhere. A key observation is if we tried to run the java server on the same port without closing the port during the first run, it would get blocked, and the only way to unblock it was to manually unblock the ports, or restart Java.

B. *Phase Two*

Phase two implementation of our project consisted of the following :

- Implementing sockets over two different devices connected over the same network.

- Setting up Android client that listens for user touch input, finger gestures which are converted into meaningful data that can be sent to the server laptop.
- Setting up the same on different networks to remotely control a laptop's keyboard and mouse.

Sockets work on the network layer. Hence controlling a remote laptop from an Android device is analogous to controlling a remote laptop from another laptop. On interpreting user touches, specific data such as the velocity of movement, time of movement, and the respective coordinate points are sent continuously to the server which indefinitely waits for client requests and handles them with the Robot class implemented in Phase One. Abstract code implemented for the same is mentioned in section 4.3

IV. LINK TO CODE

1. Server Code : http://pastebin.com/n2iAnhCM
2. Client Code : http://pastebin.com/pSEVeCtj
3. Abstract Android Code :

   http://pastebin.com/YdRX7MWQ

V. CONCLUSIONS

Mobile based devices can be used to communicate and control remote devices, given the remote device is running required code. Used ethically, mobile devices provide a powerful way of communicating with remote devices, and can be used in day to day life, in presentations during conferences or university lectures and as a substitute for IR based remote controls in televisions.

Future scope for this project could lead to research in the field of intrusion and security where study of prevention of such malicious code being injected into your device for remote control can be studied.

REFERENCES

[1] James F. Kurose, Keith W. Ross "Computer Networking: A Top Down Approach".

[2] https://docs.oracle.com/javase/tutorial/networking/sockets/

[3] unpublished.http://www.danzig.jct.ac.il/tcp-ip-lab/ibm-tutorial/3376c210.html

[4] http://www.tutorialspoint.com/java/java_networking.htm

[5] http://www.developer.com/java/other/article.php/2212401/Introduction-to-the-Java-Robot-Class-in-Java.htm